

# A Level Computer Science

## Exam Style Questions

### *Unit 1.4.2*

### *Data Structures*

### *Queues*

Name		Date	
------	--	------	--

Score	Percentage	Grade
<b>/ 38</b>		

## Question 1

- a) A program stores a queue of mathematical questions to be asked to a user. The questions are asked in the order they are added. Once a question has been asked it cannot be asked again. New questions are continually added to the end of the queue.

The program will use a non-circular queue, `questions`, (implemented using an array) to store the questions.

The pointer, `head`, stores the index of the first element in the queue.

The pointer, `tail`, stores the index of the last element in the queue.

The table below shows an example of the data in the queue.

`head` is currently 0, `tail` is currently 4.

"2*3"	"1+4"	"3-1"	"10/2"	"3+6"			
-------	-------	-------	--------	-------	--	--	--

- i. Show the contents of the queue shown above, after the following code is run

```
add("6+1")
```

--	--	--	--	--	--	--	--

[2]

- ii. State the values stored in `head` and `tail` after the code in part (i) has run.

`head`

`tail`

[2]

- b) Complete the following algorithm, to ask the user to input a new question and then either add it to the queue, or report that the queue is full.

```
procedure add()
    maxElements = 10
```

```
endprocedure
```

[4]

c) Describe why a queue is a suitable structure for this program.

[3]

d) Complete the following algorithm, to remove, and output, the first element in the queue.

procedure remove()

endprocedure

[4]

## Question 2

Stacks and queues are both data structures. A queue is shown below.

Draw what the queue would look like after the following operations:

enqueue("A"), enqueue("B"), dequeue(), enqueue("C"), dequeue(), enqueue("D")



[2]

### Question 3

The current contents of a queue, colours, implemented in an array is shown below.

red	yellow	green	blue	grey			
-----	--------	-------	------	------	--	--	--

front = 0

end = 4

The queue has the subprograms enqueue and dequeue. The subprogram enqueue is used to add items to the queue and the subprogram dequeue removes items from the queue.

a) Describe the purpose of front and end.

[2]

b) Use the following diagram to show the queue shown above after the following program statements have run:

```
enqueue("orange")
dequeue()
enqueue("maroon")
dequeue()
dequeue()
```

--	--	--	--	--	--	--	--

front =

end =

[4]

a) Describe the steps involved in the enqueue algorithm

[4]

## Question 4

The temperatures of an ocean are input into a computer system. They are recorded, and will be accessed, in the order in which they arrive. The data for one week is shown:

5, 5.5, 5, 6, 7, 6.5, 6

The data is processed. After processing, the value for the first day is stored as 0. The value for each following day is stored as an increase, or decrease, from the first day.

For example: if the first day was 7, the second was 6 and the third was 9, after processing it would be stored as 0, -1, 2.

The queue uses a function, `dequeue()` to return the first element of the queue.

Complete the algorithm to process the data in the queue and store the results in an array called `processedData`.

```
processedData[0] = 0
```

```
firstDay =
```

```
for count = 1 to 6
```

```
    processedData[ ] = dequeue() -
```

```
next count
```

[3]

## Question 5

Kamran is writing a program to manipulate the data for a set of items.  
For each item, the program needs to store:

- Item name (e.g. Box)
- Cost (e.g. 22.58)
- Date of arrival (e.g. 1/5/2018)
- Transferred (e.g. true)

The items are added to a queue for processing.  
The queue is defined as a class, `itemQueue`.

itemQueue
<code>theItems[10] : Items</code> <code>head : Integer</code> <code>tail : Integer</code> <code>numItems : Integer</code>
<code>constructor</code> <code>enqueueer()</code> <code>dequeueer()</code> <code>setnumItems()</code> <code>getnumItems()</code>

The `head` attribute points to the first element in the queue. The `tail` attribute points to the next available space in the queue. The `numItems` attribute states how many items are currently in the queue.

The array, `theItems`, stores the items in the queue. When the `tail` of the queue exceeds the last element in the array, it adds a new item to the first element if it is vacant.

For example, in the following queue, the next item to be added would be placed at index 0.

Index	0	1	2	3	4	5	6	7	8	9
Element				Data	Data	Data	Data	Data	Data	Data

- i. Define the term 'queue'.

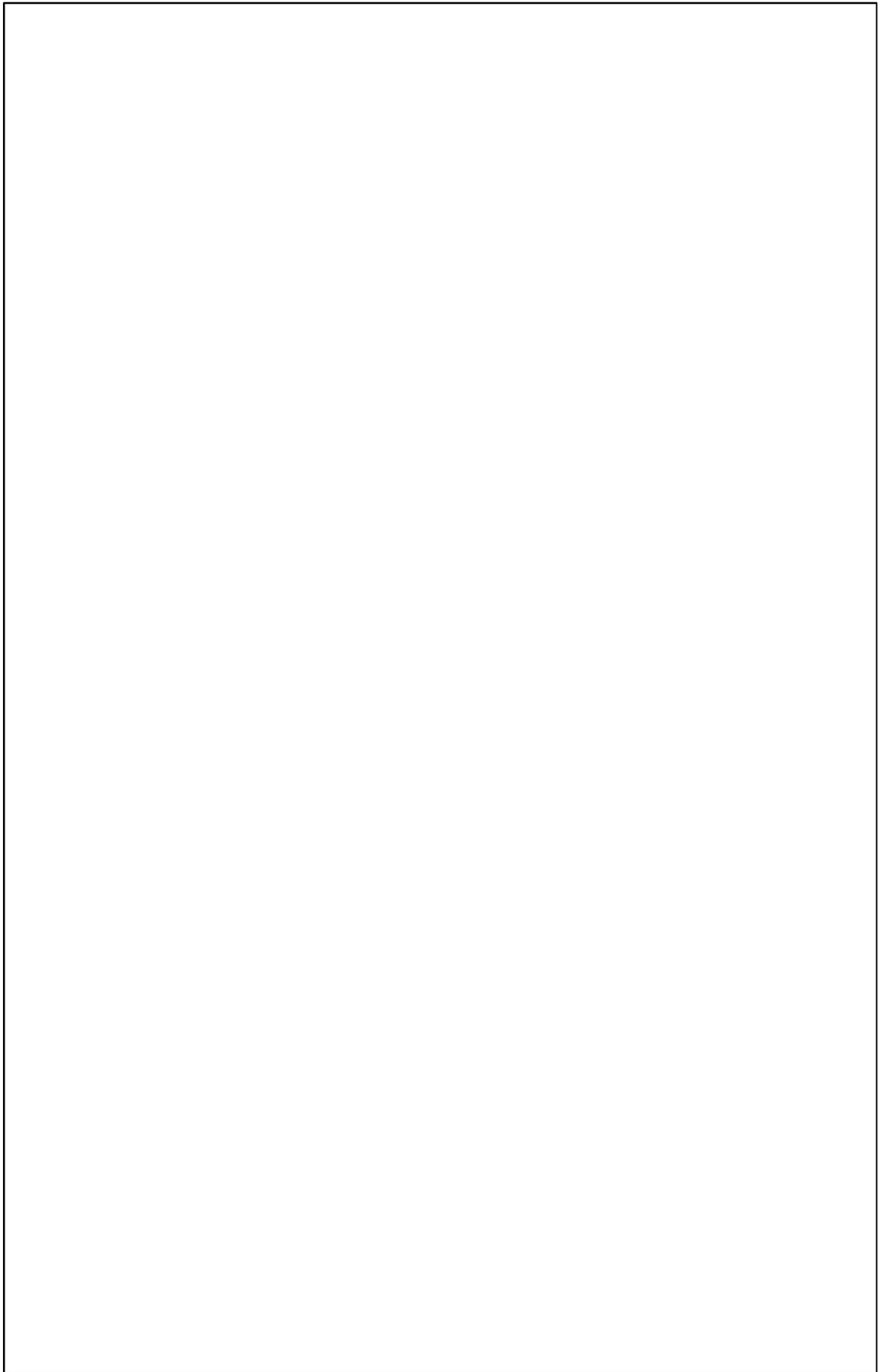
[2]

- ii. The `enqueue` method:

- takes as a parameter the item to insert in the queue
- checks if the queue is full
- reports an error and returns false if the queue is full
- does the following if the queue is not full:
  - adds the item to the array at the `tail` position and adjusts the pointer(s)
  - returns true

The attribute `numItems` stores the number of items currently in the queue.

Write an algorithm, using pseudocode or program code, for the `enqueue` method.



[6]